



Identificación de procesos: uso de algoritmos en Matlab para encontrar un modelo por identificación para un proceso de medición de pH

Pablo Parra Rosero*
e-mail: pparra@ups.edu.ec

Resumen

Se plantea la necesidad de modelar el comportamiento dinámico de un sistema de medición de pH, para lo cual podría considerarse la posibilidad de obtención de la fórmula matemática mediante parámetros físicos o mediante la identificación de un modelo matemático.

Se parte con datos obtenidos experimentalmente desde un simulador, los mismos que son generados en función de una señal PRBS, esta información de entrada y salida del simulador se utilizará para la identificación del modelo.

La identificación del modelo experimental se realiza utilizando los tres modelos de funciones paramétricas que proporciona Matlab, estos son OE, ARMAX y ARX.

Finalmente se realiza la validación cruzada de cada uno de los modelos y el análisis comparativo de los resultados para la respectiva selección del modelo matemático que mejor describa el comportamiento dinámico del sistema.

El presente trabajo no llega al cálculo del controlador del proceso, simplemente se queda en la obtención y validación del modelo.

Palabras clave: identificación, PRBS, modelo.

Abstract

This raises the need to model the dynamic behavior of a pH measurement system, for which it could be considered for obtaining the physical parameters mathematical formula or by identifying a mathematical model.

It starts with data obtained experimentally from a simulator, they are generated according to a PRBS signal, the information input and output of the simulator will be used for model identification.

The experimental model identification is performed using the three parametric models providing Matlab functions, these are: OE, ARMAX and ARX.

Finally, cross-validation is performed for each of the models and comparative analysis of the results for their selection of the mathematical model that describes the dynamic behavior of the system.

This paper does not reach the controller calculation process; simply remains in the collection and model validation.

Keywords: identification, PRBS, model

* Master en Producción Automatizada y Robótica. Ingeniero Eléctrico, Director de la Carrera de Ingeniería Eléctrica - UPS- sede Guayaquil, estudiante de Doctorado en Ingeniería con mención en automatización, control y automatización de procesos, Universidad de Piura.

Recibido: 15-julio-2011. Aprobado: 4-noviembre-2011

Forma sugerida para citar: Parra Rosero, Pablo. (2011). "Identificación de Procesos: usos de algoritmos en Matlab para encontrar un modelo por identificación para un proceso de medición de pH". *INGENIUS*. N° 6, (julio/diciembre). pp. 3-8 .ISSN: 1390-650X

1. Introducción

Las reacciones químicas y bioquímicas se desarrollan adecuadamente mientras el pH se mantiene constante; teniendo en cuenta que este tipo de reacciones ocurren tanto en seres vivos como en pruebas de laboratorio, resulta muy importante el análisis de los diferentes procesos que permiten controlar o mantener constantes los niveles de pH. Existen disoluciones amortiguadoras, conocidas también como sistemas tampón o buffers y son combinaciones sobre las cuales se pueden añadir ácidos o bases fuertes sin tener una gran variación en su concentración de protones. Las disoluciones amortiguadoras están formadas por mezclas binarias de un ácido débil y una sal del mismo ácido con una base fuerte o bien por una base débil y la sal de esta base con un ácido fuerte. [1]

En el proceso de identificación se asume que cada medición contiene disturbios, es por esta razón que para que exista la posibilidad de separar los efectos de la perturbación y las cualidades propias del sistema, resulta necesario considerar que los disturbios de variables estocásticas presentan una distribución espectral de ruido blanco y que cada medición tiene errores, los mismos que ocurren a la salida del sistema. [2]

2. Metodología

La ecuación química que modela al sistema objeto de este estudio está dada por:

$$\frac{dQ}{dt} = \frac{1}{V} (F_A C_A - 2F_B F_C - Q(F_A + F_B)) \quad (1)$$

En donde Q es la concentración final. Para encontrar el pH en función de Q (en cada punto), se relacionan las ecuaciones que definen el comportamiento de la reacción ácido fuerte-base débil, obteniendo:

$$Q = 10^{-pH} - \frac{2K_W}{10^{-pH}} \quad (2)$$

De donde puede obtenerse la ecuación para pH:

$$pH = -\text{Log}[\sqrt{0.25 * Q^2 + 2 * K_W} + 0.5 * Q] \quad (3)$$

2.1 Diseño del experimento

Para la simulación se han utilizado distintas entradas escalón (en un intervalo de tiempo cada una), con el fin de observar el comportamiento de la salida pH según diferentes entradas de caudal de base.

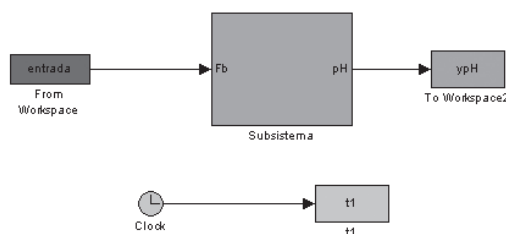


Fig. 1. Diagrama de bloques del sistema de simulación

Para las pruebas en el laboratorio se ha considerado un pH máximo de 3,5. En la figura 1 se muestra el diagrama de bloques que representa el sistema utilizado para la generación de la señal PRBS (pseudo-random-binary-signals). En la figura 2 se observa el algoritmo utilizado para ejecutar la ecuación diferencial que representa la variación de Q con respecto al tiempo.

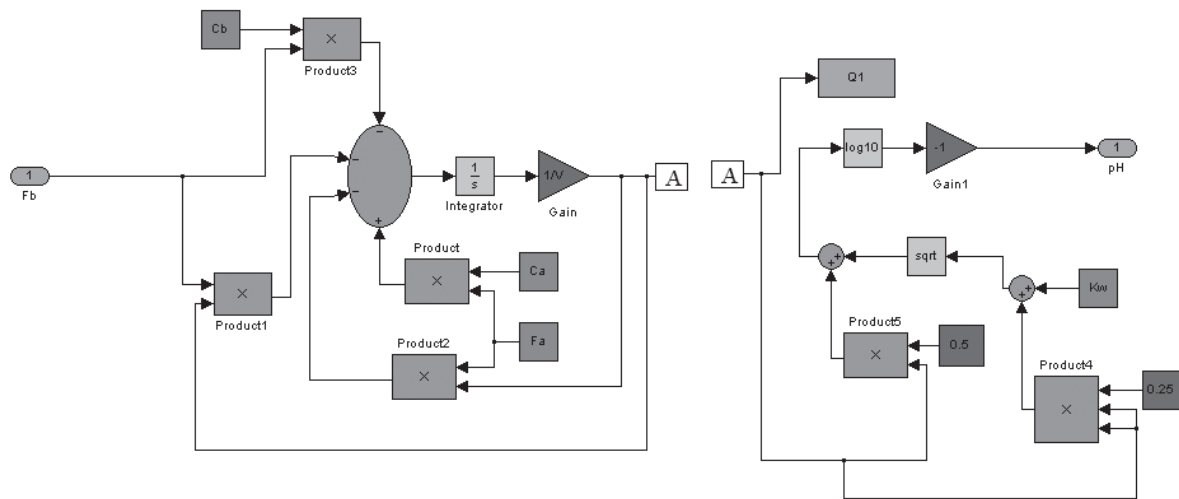


Fig. 2. Algoritmo para la ecuación (1)

2.2 Identificación

Para el proceso de identificación se aplica una señal PRBS considerando los siguientes parámetros:

- Caudal de base nominal: 0,004167 L/s
- pH máximo: 2,94

La PRBS (figura 3) se obtiene ejecutando un algoritmo, el cual toma en cuenta parámetros como:

- Tiempo de muestreo: 5 (para señal PRBS)
- Tiempo de simulación: 15*1200
- Valor nominal de la variable manipulable: 0,004167
- Tiempo mínimo de conmutación de la señal: 200
- Valores máximos y mínimos de la señal respecto a la referencia: 20%

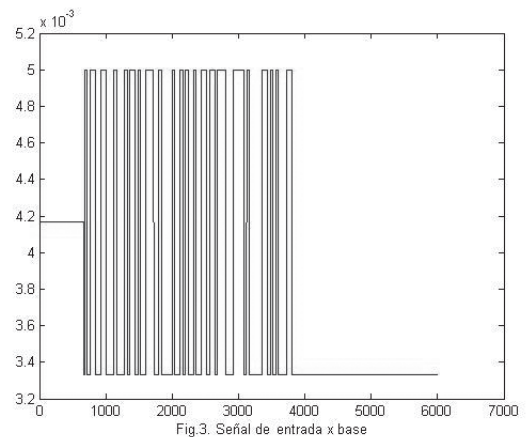


Fig.3. Señal de entrada x base

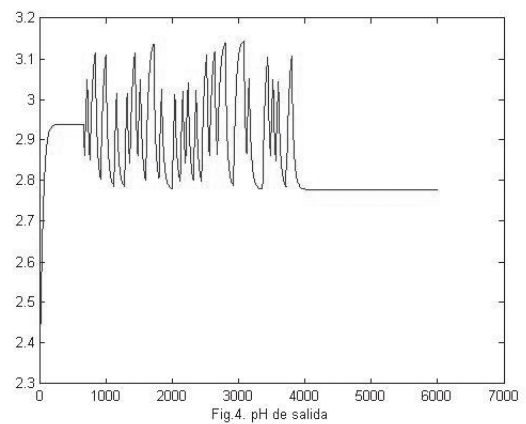


Fig.4. pH de salida

2.2 Identificación

En las figuras 3 y 4 se puede apreciar que las señales de entrada y salida no varían sino a partir de 330 segundos aproximadamente; además, la señal de entrada tiene su variación alrededor de $4,2 \times 10^{-3}$ y la señal de salida alrededor de 2,9. Esto implica que se debe realizar un proceso de pre-blanqueo de los datos.

Además deben dividirse los datos en dos secciones, una parte de los datos será utilizada para la obtención del modelo y la otra para la validación del mismo.

3. Resultados

Utilizando el Matlab, en z se almacena el arreglo de datos de entrada y salida, pre-blanqueados correspondientes a la porción que se utilizará para la obtención del modelo. [3]

3.1 Encontrando el modelo ARX

```
>> m = arx(z,[2 2 1])
```

Discrete-time IDPOLY model: $A(q)y(t) = B(q)u(t) + e(t)$
 $A(q) = 1 - 1,864 q^{-1} + 0,8673 q^{-2}$

$B(q) = 6,488 q^{-1} - 5,825 q^{-2}$

Estimated using ARX from data set z
 Loss function 9,08695e-008 and FPE 9,13325e-008
 Sampling interval: 1

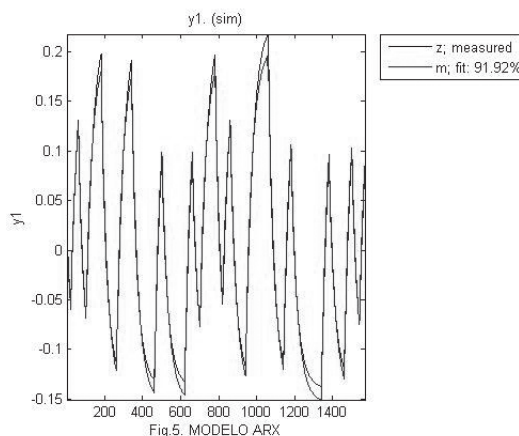


Fig.5. MODELO ARX

En la figura 5 se aprecia la gráfica que compara el comportamiento real del sistema (datos simulados) con el modelo ARX obtenido. Se alcanza un fit del 91,92%; este parámetro mide el porcentaje de similitud entre las dos señales.

Se procede a validar el modelo ARX obtenido (m), comparando dicho modelo con la porción de datos reservada para la validación cruzada «zv». En la figura 6 se observa la comparación de las gráficas obtenidas.

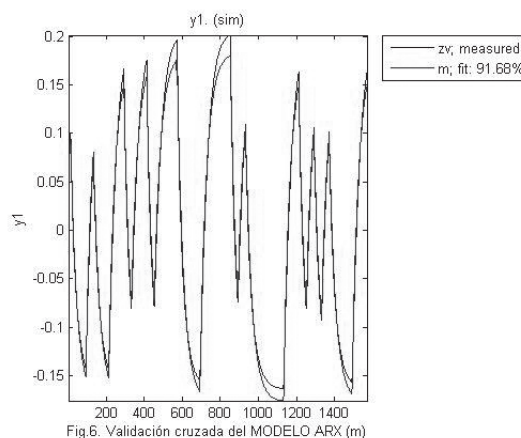


Fig.6. Validación cruzada del MODELO ARX (m)

3.2 Encontrando el modelo ARMAX

```
>> m1 = armax(z,[2 2 1 1])
```

Discrete-time IDPOLY model: $A(q)y(t) = B(q)u(t) + C(q)e(t)$

$A(q) = 1 - 1,861 q^{-1} + 0,8642 q^{-2}$

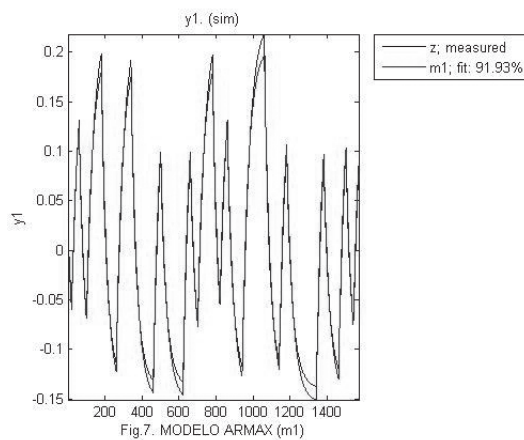
$$B(q) = 6,488 q^{-1} - 5,806 q^{-2}$$

$$C(q) = 1 + 0,01593 q^{-1}$$

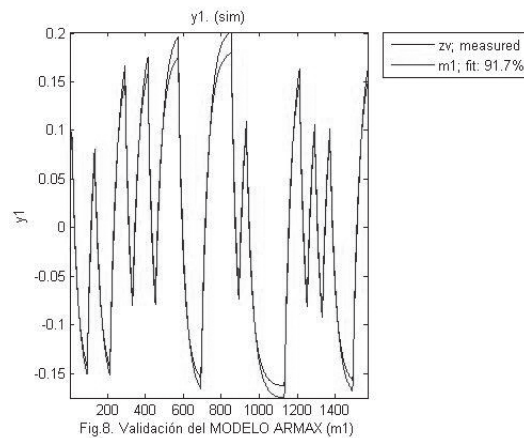
Estimated using ARMAX from data set z
Loss function 9,09663e-008 and FPE 9,15464e-008

Sampling interval: 1

En la figura 7 se aprecia la comparación entre el modelo «m1» ARMAX y los datos medidos.



Un fit del 91,92% refleja aparentemente un mejor comportamiento frente a los resultados obtenidos con el modelo «m» ARX.



Utilizando el arreglo de datos zv, se procede a validar el modelo «m1» ARMAX, el fit del 91,71% refleja un gran nivel de similitud con los datos reales.

3.3 Encontrando el modelo OE (Output Error)

```
>> moe = oe(z,[2 4 1])
```

Discrete-time IDPOLY model: $y(t) = [B(q)/F(q)]u(t) + e(t)$

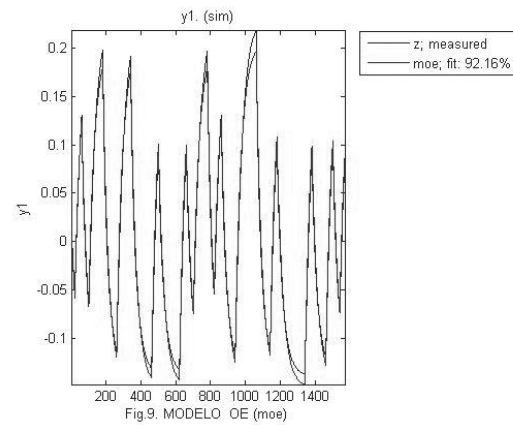
$$B(q) = 6,425 q^{-1} - 6,424 q^{-2}$$

$$F(q) = 1 - 1,967 q^{-1} + 0,9851 q^{-2} - 0,03858 q^{-3} + 0,02062 q^{-4}$$

Estimated using OE from data set z

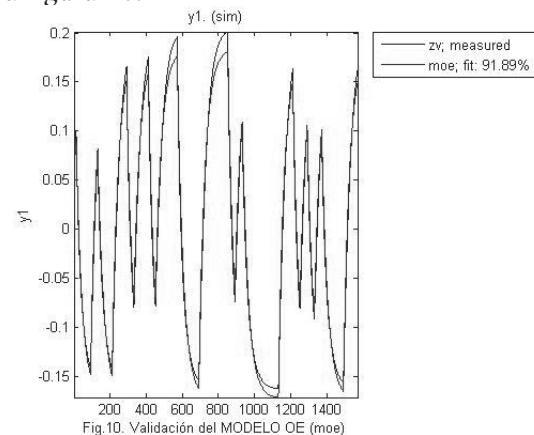
Loss function 6,49396e-005 and FPE 6,54385e-005

Sampling interval: 1



Para este modelo se puede notar una mejor aproximación, puesto que se supera el 92% en el índice fit.

En la validación cruzada de igual manera se obtiene un fit del 91,89%, como se puede observar en la gráfica comparativa mostrada en la figura 10.



4. Discusión

Los modelos obtenidos y debidamente validados representan el funcionamiento dinámico del sistema con gran aproximación, puesto que se ha conseguido que los índices fit para los tres casos OE, ARMAX y ARX, sean muy superiores al 80%, que es un índice referencial de buen comportamiento en este tipo de modelamiento. Observe figura 11.

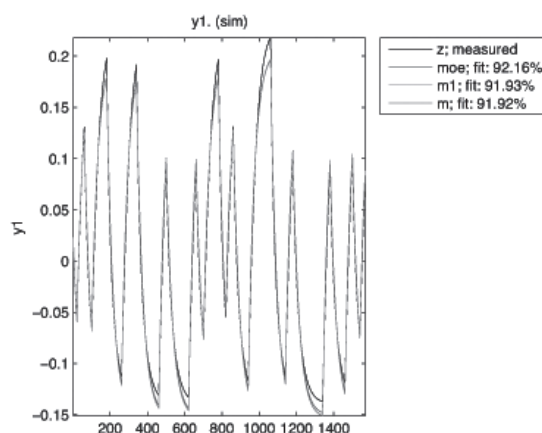


Fig. 11. Comparación de MODELOS ARX(m), ARMAX(m1) y OE(moe)

El número de parámetros constantes encontrados para los distintos modelos no excede de 6, lo que indica que estos modelos son sencillos de simular y procesar matemáticamente.

La selección de modelos puede realizarse por ejemplo, según Akaike, considerando el menor valor de FPE (error final de predicción) o el AIC (criterio final de Akaike) y el índice de la función de pérdida *Loss function*.

En la tabla 1 se contrastan los valores de los índices FPE y *Loss function* para cada uno de los modelos encontrados:

Tabla 1. Índices FPE y *Loss function*

	ARX	ARMAX	OE
Loss function	9,08695e-8	9,09663e-8	6,49396e-5
FPE	9,13325e-8	9,15464e-8	6,54385e-5

Analizados los datos, se puede concluir que el modelo ARX «m» muestra una mejor performance.

5. Conclusiones

Utilizando las herramientas proporcionadas por el *software* Matlab, resulta sencillo obtener los distintos modelos y validarlos. Los resultados obtenidos permiten afirmar que dichos modelos se aproximan de excelente manera al comportamiento real del sistema en estudio.

Resultará de gran interés diseñar o calcular un controlador para este sistema, considerando uno de los modelos obtenidos y luego implementar el mismo de forma real, para analizar o constatar la efectividad de este proceso de identificación para aplicación en sistemas de control automático.

Referencias bibliográficas

- [1] EHU. **Regulación del pH – Amortiguadores**. Universidad del País Vasco pdf/5_buffers.p
- [2] Ipanaqué W., Del Ré L. 2011. **Elementos básicos de identificación**. Separata DICOP, mayo.
- [3] Ljung L. 1988. **Matlab User's Guide: System Identification Toolbox**.